(approximate search) Given a nonempty sorted list of numbers and a number, write a program to determine the index of an item that is closest in value to the given number.

After trying the question, scroll down to the solution.
Linear search is easiest, but the list is sorted, so let's make it a binary search. Let the list be \( L \), let the number be \( x \), and let the natural variable whose final value will be the index of a closest value be \( h \). We also need natural variables \( i \) and \( j \). The problem specification \( R \), and specification \( Q \), are defined as

\[
R = \text{abs} \ (x - L \ h') = (\ll k: 0,..\#L: \text{abs} \ (x - L \ k))
\]

\[
Q = \text{abs} \ (x - L \ h') = (\ll k: h,..j: \text{abs} \ (x - L \ k))
\]

Here are the refinements.

\[
R \iff h := 0. \ j := \#L. \ j - h \geq 1 \Rightarrow Q
\]

\[
j - h \geq 1 \Rightarrow Q \iff \text{if } j - h = 1 \text{ then } j - h = 1 \Rightarrow Q \text{ else } j - h > 1 \Rightarrow Q \text{ fi}
\]

\[
j - h = 1 \Rightarrow Q \iff \text{ok}
\]

\[
j - h > 1 \Rightarrow Q \iff j - h > 1 \Rightarrow h' = h < i' < j' \colon
\]

\[
\text{if } \text{abs} \ (x - L (i - 1)) > \text{abs} \ (x - L \ i) \text{ then } h := i \text{ else } j := i \text{ fi}
\]

\[
j - h \geq 1 \Rightarrow Q
\]

The comparison \( \text{abs} \ (x - L (i - 1)) > \text{abs} \ (x - L \ i) \) could just as well have been \( \text{abs} \ (x - L (i - 1)) \geq \text{abs} \ (x - L \ i) \).

\[
j - h > 1 \Rightarrow h' = h < i' < j' \iff i := \text{div} \ (h + j) \ 2
\]

For getting the right result, it doesn't matter where \( i' \) is, as long as \( h < i' < j' \). But for the time bound, it does matter that \( i' \) is halfway between \( h \) and \( j \).

Proof of the \( R \) refinement, starting with the right side.

\[
\begin{align*}
h & := 0. \ j := \#L. \ j - h \geq 1 \Rightarrow Q \\
& = h := 0. \ j := \#L. \ j - h \geq 1 \Rightarrow \text{abs} \ (x - L \ h') = (\ll k: h,..j: \text{abs} \ (x - L \ k)) \\
& = \text{abs} \ (x - L \ h') = (\ll k: 0,..\#L: \text{abs} \ (x - L \ k)) \\
& = R
\end{align*}
\]

Proof of the \( j - h \geq 1 \Rightarrow Q \) refinement, by cases. First case:

\[
\begin{align*}
& j - h = 1 \land (j - h = 1 \Rightarrow Q) \\
& = j - h = 1 \land Q \\
& \Rightarrow Q \\
& \Rightarrow \neg j - h \geq 1 \lor Q \\
& = j - h \geq 1 \Rightarrow Q
\end{align*}
\]

Last case:

\[
\begin{align*}
& j - h + 1 \land (j - h + 1 \Rightarrow Q) \Rightarrow (j - h \geq 1 \Rightarrow Q) \\
& = j - h + 1 \land (j - h + 1 \Rightarrow Q) \land j - h \geq 1 \Rightarrow Q \\
& = j - h + 1 \land (j - h + 1 \Rightarrow Q) \Rightarrow Q \\
& = j - h + 1 \land Q \Rightarrow Q \\
& = \top
\end{align*}
\]

Proof of \( j - h = 1 \Rightarrow Q \) refinement:
I have not done the timing, but it’s a binary search.

Proof of \( j-h>1 \Rightarrow Q \) refinement, staring with the right side:

\[
\begin{align*}
j-h>1 & \Rightarrow h'=h<i'<j'. \\
\text{if} \ abs \ (x-L(i-1)) > abs \ (x-L \ i) & \text{then} \ h:=i \ \text{else} \ j:=i \ \text{fi}.
\end{align*}
\]

\[
\begin{align*}
j-h\geq 1 & \Rightarrow Q \\
\text{distribute} \ j-h\geq 1 \Rightarrow Q
\end{align*}
\]

\[
\begin{align*}
j-h>1 & \Rightarrow h'=h<i'<j'. \\
\text{if} \ abs \ (x-L(i-1)) > abs \ (x-L \ i) & \text{then} \ h:=i \ \text{else} \ j:=i \ \text{fi}.
\end{align*}
\]

\[
\begin{align*}
j-h\geq 1 & \Rightarrow Q
\end{align*}
\]

Proof of the \( j-h>1 \Rightarrow h'=h<i'<j'=j' \) refinement:

\[
\begin{align*}
(j-h>1 \Rightarrow h'=h<i'<j') & \iff i:= \text{div} \ (h+j) \ 2 \\
\text{portation}
\end{align*}
\]

\[
\begin{align*}
(j-h>1 \land (i:= \text{div} \ (h+j) \ 2)) & \Rightarrow h'=h<i'<j'
\end{align*}
\]

\[
\begin{align*}
(j-h>1 \land h'=h \land i'= \text{div} \ (h+j) \ 2 \land j'=j) & \Rightarrow h'=h<i'<j'
\end{align*}
\]

\[
\begin{align*}
\top
\end{align*}
\]

I apologize for the informality of this step. Since \( i \) is properly between \( h \) and \( j \), both \( L(i-1) \) and \( L \ i \) are in the interval \( h,.j \). After the \( \text{if} \), the comparison says that \( x \) is closer in value to \( L \ i \) than to \( L(i-1) \). Since the list is sorted, that means \( x \) is closer in value to some item in \( i,.j \) than to any item in \( h,.i \). So the closest item in the whole interval \( h,.j \) is to be found in the right part \( i,.j \). Likewise if the comparison is false, the closest item in the whole interval \( h,.j \) is to be found in the left part \( h,.i \).

\[
\begin{align*}
\Rightarrow j-h>1 & \Rightarrow abs \ (x-L \ h') = (\downarrow k: h,.j: \ abs \ (x-L \ k)) \\
\text{definition of} \ Q
\end{align*}
\]

\[
\begin{align*}
j-h>1 & \Rightarrow Q
\end{align*}
\]